IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 January 2024, accepted 29 January 2024, date of publication 5 February 2024, date of current version 9 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3362677

RESEARCH ARTICLE

Neighborhood Ranking-Based Feature Selection

ÁDÁM IPKOVICH AND JÁNOS ABONYI
HUN-REN-PE Complex Systems Monitoring Research Group, University of Pannonia, 8200 Veszprém, Hungary
Corresponding author: János Abonyi (janos@abonyilab.com)

This work was supported by the TKP2021-NVA-10 Project through the Ministry for Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the 2021 Thematic Excellence Programme Funding Scheme and the Hungarian Research Fund, under Grant OTKA 143482 (Monitoring Complex Systems by Goal-Oriented Clustering Algorithms). The work of Ádám Ipkovich was supported by the ÚNKP-22-1 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

ABSTRACT This article aims to integrate *k*-NN regression, false-nearest neighborhood (FNN), and trustworthiness and continuity (T&C) neighborhood-based measures into an efficient and robust feature selection method to support the identification of nonlinear regression models. The proposed neighborhood ranking-based feature selection technique (NRFS) is validated in three problems, in a linear regression task, in the nonlinear Friedman database, and in the problem of determining the order of nonlinear dynamical models. A neural network is also identified to validate the resulting feature sets. The analysis of the distance correlation also confirms that the method is capable of exploring the nonlinear correlation structure of complex systems. The results illustrate that the proposed NRFS method can select relevant variables for nonlinear regression models.

INDEX TERMS Machine learning, nonlinear regression, feature selection, k-nearest neighbors, model-free regression, trustworthiness and continuity, distance correlation.

I. INTRODUCTION

Neighborhood-based methods are model-free, nonparametric algorithms that excel in feature selection tasks due to the lack of costly model identification and evaluation.

The *k*-nearest neighbors (*k*-NN) method is a staple for solving regression and classification problems due to its lazy evaluation. During regression, the objective is to approximate the output by computing the mean of the dependent variables associated with the *k* neighbors of the point in the independent variables. Therefore, the method skips the model identification process and contains only one hyperparameter that is often determined by the data structure. In regression-based feature selection, the prediction error may decrease significantly if only the relevant variables remain during the selection process. As such, a model-free nonparametric feature selection can be performed with the help of the *k*-NN.

One of the key ideas of this work is that if *k*-NN is capable of feature selection, other similar methods may also be capable of it. Therefore, several neighborhood-based methods have been examined and we have established a connection between them. An example is the false-nearest neighbors (FNN) method, single closest neighbor case of *k*-NN that is also capable of feature selection.

The FNN method compares the data with its closest neighbor. The relationship of distances in the independent and dependent variables determines whether it is a false neighbor [1] by comparing the steepness between the points with a threshold hyperparameter. The number of neighbors that are above a threshold value defines the quality of the relationship. The threshold can be determined based on the Jacobian matrix of the data [2]. In a sense, the data contain threshold values, which can be estimated from the local covariance matrix. The FNN sums up the number of false neighbors that are above the threshold and divides it by the total number of data, resulting in a single score. The number of false neighbors measures the degree of correlation of the variables, which is supported by the ability of the method to identify model structures [2]. FNN has been used primarily to determine embedding dimensions, however, another neighborhood-based method, namely trustworthiness and continuity (T&C), is capable of that as well.

The associate editor coordinating the review of this manuscript and approving it for publication was Jethro Browell.

The T&C method focuses on the one-way embedding of a set of variables to another, which can determine the correlation [3]. Trustworthiness quantifies the projection of the dependent variables against the independent variables by establishing the local neighborhood in the dependent variable and finding which neighborhoods cannot be considered the same in the independent variables. Continuity can be regarded as the measurement of the projection quality of the independent variables to the dependent variables. Other works featuring the T&C for feature selection are unknown to the authors.

The connection between the methods has not yet been established in a compiled work, and while k-NN and FNN have been used with or as feature selection methods, T&C has not yet been labeled as a feature selection method in the literature. Moreover, as there are no methods that are capable of solving all problems, neighborhood-based techniques can be used somewhat interchangeably due to their similar model-free, nonparametric nature. As the authors were unable to find scientific work about the connections between neighborhood-based methods and their collective integration into a regression-based feature selection framework, the article aims to integrate *k*-NN, FNN, and T&C neighborhood-based methods into a novel approach to robust feature selection of nonlinear models. The methods are model-free and nonparametric, and therefore, the runtime required to perform feature selection is reduced. Each core neighborhood-based method is included in the framework, including *k*-NN, FNN, T&C, and a novel neighborhood ranking-based feature selection algorithm.

Inspired by neighborhood-based methods, we incorporate novel local neighborhood ranking-based methods for model-free feature selection. The key idea is that the rankings of the independent and dependent variables differ as the ordering of the neighbors may change according to the model, and therefore their differences may provide the necessary information about the relevance of a variable. If the ranking difference of the points in the neighborhoods is substantial, then the relationship may be false or non-existent and is capable of qualifying nonlinear relationships as well.

The proposed Neighborhood Ranking-Based Feature Selection (NRFS) method ranks the distance matrices of both variables, after which the ranking differences of the matrices are taken and summed similarly to the Sum of Ranking Differences technique [4], [5], [6]. During the examination of the local neighborhood, NRFS can be localized to measure the one-way correlation by subtracting the local ranks from the independent variable to the corresponding ranks of the dependent variable (NRFSX) or *vice versa* (NRFSY). These ''local'' variations also consider a one-way nearest-neighbor rank evaluation similar to the T&C. The connection between the two becomes evident, though the total difference between the rankings (NRFS) is not able to evaluate the nonlinear correlation. According to the benchmarks, the NRFSX (measuring the correlation of the independent to the dependent) selected the features the fastest while retaining precision similar to the continuity measure.

The techniques mentioned above can be used to evaluate the nonlinear correlation between independent and dependent variables. However, these neighborhood-based methods do not actively select features but only measure the correctness of the combination. Therefore, we use the techniques as cost functions to optimize algorithms such as brute force, forward selection [7] and genetic algorithms [8]. The methods are validated against distance correlation, distance rank correlation, and neural network-based feature selection.

As such, this paper focuses on regression-based feature selection of neighborhood-based methods to eliminate model identification and tackle nonlinear correlation. The algorithms are tested on three distinct datasets, and are benchmarked against each other. Therefore, the contributions of this work can be defined as follows.

- First, we define the FNN as the special case of *k*-NN, where the number of neighbors is defined as one. With leave-one-out validation, we can determine a threshold value that can be used to measure nonlinear correlation robustly without model specificity. Second, we interpret FNN as the special case of continuity metrics, where the pairwise connection between two variables is measured. Moreover, we propose a model-free feature selection method based on the FNN and rank correlation of local groups to select the relevant variables and determine the correlation or causality of these variables.
- We employ brute force, forward selection, and genetic algorithms to select the correct variation of the features by incorporating neighborhood-based methods as cost functions into optimization.
- We benchmark on a dynamic modeling example, proposing that the methods can select the order of a dynamic model. We also benchmark on a simple and widely used dataset, the Friedman dataset [9].
- We introduce the theoretical background for the false *k*-nearest neighbors (F*k*-NN) technique, which is to generalize the false nearest neighbors for the *k*-nearest neighbors.

The following Section II (The method of neighborhood ranking-based feature selection) discusses the algorithms. We first introduce the theoretical background of neighborhood-based methods in Section II-A (The background of neighbor-based methods), followed by the definition of the *k*-NN algorithm (Section II-B: *k*-nearest neighbors with leave-one-out regression) and its special case, FNN (Section II-C: The False nearest neighbors method). Then T&C is described in Section II-D (Trustworthiness and continuity) before defining the generalized NRFS method (Section II-E: Neighborhood ranking-based feature selection). The connections and generalizations between the methods are discussed in Section II-F (Discussion on the similarities of the measures). The related works are introduced after the discussion (Section II-G: Related works of neighborhood-

based methods). The methods are applied to three datasets: Simple linear, monotonous, and nonlinear equations are examined first in section III-A: Linear, monotonous, and periodic functions. The Friedman-1 dataset is evaluated second(Section III-B: Friedman-1 model), in which we describe the optimal number of neighbors for the dataset and the use of forward selection and genetic algorithms to execute feature selection. A neural network is also applied to validate the results. Lastly, the identification of dynamic models is described in Section III-C: Dynamic modeling - polymerization reactor, and the work is concluded in the Conclusion (Section IV).

II. THE METHOD OF NEIGHBORHOOD-RANKING-BASED FEATURE SELECTION

Regression models assume a functional relationship $y_i = f(\mathbf{x}_i)$, where the values of n independent variables can be denoted as $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]$). Models can predict more than one dependent variable (y_i), for simplicity, this work considers only one output. Prediction error increases with each irrelevant input added to the set of independent variables, as they do not provide a substantive contribution to the dependent variable. For this reason, selecting the relevant features is essential to determine a functioning model. Model-free machine learning methods assume a black-box approach and, therefore, do not require parameters or the mathematical background of the model. Neighborhood-based methods are a subset of model-free learning and often establish the connection between variables properly without proper interpretation. Model-based algorithms often reflect the concept in an interpretable way, but they are costly in both time and computational power.

The feature selection problem can be formalized with an n number of independent (input) variables, each with N observations $\mathbf{X}_{N\times n}$. The models estimate the dependent (output) variables $\mathbf{y} = [y_1, y_2, \ldots, y_N]$ for each point in the independent variable space. The objective of the article is to use neighborhood-based algorithms to select features, as they provide favorable predictions compared to the original value when the appropriate variables are included.

A. THE BACKGROUND OF NEIGHBOR-BASED METHODS

This section establishes the basic definitions required throughout the paper. Each neighborhood-based method evaluates the local environment of a point based on the number of closest neighbors, which can be established based on the Euclidean (L^2-norm) distance between the points. As such, for the mth point and the ith point, the distance function can be defined as follows:

$$\mathrm{d}(\mathbf{x}_i, \mathbf{x}_m) = ||\mathbf{x}_i - \mathbf{x}_m||_2;\ i = 1, \ldots, N;\ m = 1, \ldots, N, \tag{1}$$

where d() denotes the Euclidean distance between the two points, m represents the running index of a point whose distance is calculated against the ith point.

A distance matrix represents the pairwise distances of all points.

$$\mathbf{D}^x = \begin{bmatrix} \mathrm{d}(\mathbf{x}_1, \mathbf{x}_1) & \ldots & \mathrm{d}(\mathbf{x}_N, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \mathrm{d}(\mathbf{x}_1, \mathbf{x}_N) & \ldots & \mathrm{d}(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{2}$$

$$\mathbf{D}^y = \begin{bmatrix} \mathrm{d}(y_1, y_1) & \ldots & \mathrm{d}(y_N, y_1) \\ \vdots & \ddots & \vdots \\ \mathrm{d}(y_1, y_N) & \ldots & \mathrm{d}(y_N, y_N) \end{bmatrix} \tag{3}$$

It is important to note that the distance matrices of the independent and dependent variables ($\mathbf{D}^x$ and $\mathbf{D}^y$) are symmetric so that $\mathrm{d}(\mathbf{x}_m, \mathbf{x}_i) = \mathrm{d}(\mathbf{x}_i, \mathbf{x}_m)$.

The neighbor rank with regards to the ith point can be defined based on ith column of the distance matrix. The rank determines which points are closest to the ith point.

$$r^x_{i,m} = \mathrm{rank}(\mathrm{d}(\mathbf{x}_i, \mathbf{x}_m)),\ r^x_{i,m} \neq r^x_{m,i} \tag{4}$$

where $r^x_{i,m}$ is the rank value of the mth point to the ith point according to the ranking function rank().

A rank matrix $\mathbf{R}$ can be constructed similarly to the distance matrix ($\mathbf{D}$).

$$\mathbf{R}^x = \begin{bmatrix} r^x_{1,1} & \cdots & r^x_{N,1} \\ \vdots & \ddots & \vdots \\ r^x_{1,N} & \cdots & r^x_{N,N} \end{bmatrix}$$
$$r^x_{i,m} \neq r^x_{m,i},\ i, m \in 1, \ldots, N;\ i \neq m \tag{5}$$

where $\mathbf{R}^x$ denotes the ranking matrix.

The ranking matrix is the ordinality of the distance matrix. Here, the ranking function ranks columnwise, therefore, row vectors cannot be considered a coherent ranking vector.

As such, the mth data can be considered the jth closest neighbor of the ith point if:

$$i_x(j) = \left\{ m \in \{1, \ldots, N\} \middle| r^x_{i,m} = j \right\}$$
$$j = 1, \ldots, N;\ m \neq i \tag{6}$$

where $i_x(j)$ denotes the jth neighbor of the ith point in the independent variables.

If the distance rank is less than or equal to k, then the point $i_x(j)$ is the kth neighbor of the ith point. The set of neighbor indices $S_k(\mathbf{x}_i)$ can be defined as follows:

$$S_k(\mathbf{x}_i) = \left\{ i_x(j) \in \{1, \ldots, N\} \middle| r^x_{i,i_x(j)} \leq k \right\} \tag{7}$$

where $S_k(\mathbf{x}_i)$ denotes the set of the kth closest neighbors according to the ith point.

Neighborhood-based methods use nearest neighbors for regression, classification, or correlation measurement; an example is the k-nearest neighbors (k-NN) algorithm [10].

B. K-NEAREST NEIGHBORS WITH LEAVE-ONE-OUT REGRESSION

In k-nearest neighbors (k-NN)-based lazy regression with leave-one-out validation, for each point ($\mathbf{x}_i$), the output y_i is estimated by finding the k-nearest neighbors in the set of independent variables and aggregating the corresponding weighted values in the dependent variables [10].

As such, the predicted output $\hat{y}_i$ at $\mathbf{x}_i$ can be calculated by taking the mean of the outputs of the neighbors:

$$\hat{y}_i = \frac{1}{k} \sum_{i_x(j) \in S_k(\mathbf{x}_i)} w_{i_x(j)} y_{i_x(j)} \tag{8}$$

where $\hat{y}_i$ denotes the predicted value of the dependent variable, $y_{i_x(j)}$ represents the jth neighbor of the ith point in the independent variables. $w_{i_x(j)}$ stands for the weight of $y_{i_x(j)}$.

The principle behind k-NN is illustrated in Figure 1. The algorithm calculates the Euclidean distance in the independent variables, finds the kth closest neighbor with the least distances, and takes the mean of their values in the dependent variable.

The neighbors can be assigned an equal weight $1/k$ or their similarity to the selected value [11]:

$$w_{i_x(j)} = \frac{\frac{1}{\mathrm{d}(\mathbf{x}_i, \mathbf{x}_{i_x(j)})}}{\frac{1}{\sum_{i(j) \in S_k(\mathbf{x}_i)} \mathrm{d}(\mathbf{x}_i, \mathbf{x}_{i_x(j)})}} \tag{9}$$

The mean squared error of the prediction defines how accurate the k-NN regression is. The error of one point to its neighbor is as follows:

$$\begin{aligned} e_i &= \|y_i - \hat{y}_i\|_2 = \\ &= \left\| y_i - \frac{1}{k} \sum_{i_x(j) \in S_k(\mathbf{x}_i)} w_{i_x(j)} y_{i_x(j)} \right\|_2 \Bigg| i_x(j) \in S_k(\mathbf{x}_i) \\ \epsilon &= \frac{1}{N} \sum_{i=1}^{N} e_i \end{aligned} \tag{10}$$

where e_i denotes the squared error of the predicted data $\hat{y}_i$ and the original data y_i. ϵ represents the mean squared error of the predicted and original data.

k-NN is sensitive to features. The error may increase drastically if the set contains irrelevant features. Thus, it can be used for feature selection; the lower the error value (ϵ), the better the current combination of features may become.

C. THE FALSE NEAREST NEIGHBORS METHOD

The false nearest neighbor (FNN) technique was created to determine the minimum embedding dimension of the models [1]. The method was later applied to successfully analyze the relationship between the inputs and outputs of the models [12]. The FNN method shares a connection with the k-NN technique, where the evaluation of a point requires its closest neighbor ($k = 1$) in the independent variables. The distances of two points are calculated in both sets of variables, whose quotients are compared to a threshold value. A point has a false neighbor if the quotient is above the threshold. The ratio of false neighbors to all points determines the quality of the projection.

Let us suppose that there is a connection between the independent and dependent variables, which can be modeled as [13]:

$$y_i - y_{i_x(1)} \approx \sum_{l=1}^{n} \frac{\partial \mathrm{f}}{\partial x_{i,l}} (x_{i,l} - x_{i_x(1),l}) \tag{11}$$

where $y_i - y_{i_x(1)}$ denotes the change in the dependent variable, $x_{i,l}$ represents the ith observation of the lth variable and $\frac{\partial \mathrm{f}}{\partial x_{i,l}}$ stands for the partial derivative of the underlying model; in other words, $y_i - y_{i_x(1)}$ is approximated as the first order Taylor series of model f(). Thus, FNN is based on the linearization of f() around the point, including the first neighbor $i_x(1)$.

With the help of the Cauchy-Schwarz inequality, Eq. 11 can be reorganized [2]:

$$|y_i - y_{i_x(1)}| \leq \left\| \frac{\partial \mathrm{f}}{\partial \mathbf{x}_i} \right\|_2 \|\mathbf{x} - \mathbf{x}_{i(1)}\|_2 \tag{12}$$

$$\frac{|y_i - y_{i_x(1)}|}{\|\mathbf{x}_i - \mathbf{x}_{i(1)}\|_2} \leq \left\| \frac{\partial \mathrm{f}}{\partial \mathbf{x}_i} \right\|_2 = \alpha \tag{13}$$

If this inequality is true, the nearest neighbor is a good neighbor. The selection of α is both an essential and gruesome task, as it is impossible to select a robust threshold for all databases. The threshold value directly influences the determination of false neighbors (neighbors that are not good neighbors); thus, overestimating it may deteriorate the accuracy of the method. The threshold can be predicted based on the Jacobian matrix. In a practical sense, this value can be estimated based on the (local) covariance matrix of the data.

The FNN algorithm examines the relationship between the value of the closest neighbor in the independent variables and the value of its corresponding dependent variable. The more false neighbors there are, the worse the relationship between the input and the output is. The set of points or samples with false neighbors can be defined as:

$$F_1(\mathbf{x}_i, y_i) = \left\{ \frac{||y_i - y_{i_x(1)}||_2}{||\mathbf{x}_i - \mathbf{x}_{i_x(1)}||_2} > \alpha \Bigg| i_x(1) \in S_1(\mathbf{x}_i) \right\} \tag{14}$$

where the set $F_1(\mathbf{x}_i, y_i)$ denotes whether the nearest neighbor is located in the false neighbor set.

The inequality determines the neighbor. Neighborhoods are asymmetric, as a neighbor in one set of variables may not have the same rank in the other, which may indicate a false or one-way relationship. The more false neighbors there are, the worse the accuracy of the model.

$$\mathrm{FNN} = 100 \frac{1}{N} \sum_{i=1}^{N} |F_1(\mathbf{x}_i, y_i)| \tag{15}$$

where the length of the set of nearest neighbor $|F_1(\mathbf{x}_i, y_i)|$ is at maximum one.

Figure 2 depicts the nearest neighbor and false nearest neighbor algorithms. Figure A) illustrates the k-NN algorithm with $k = 1$. Here, leave-one-out cross-validation is

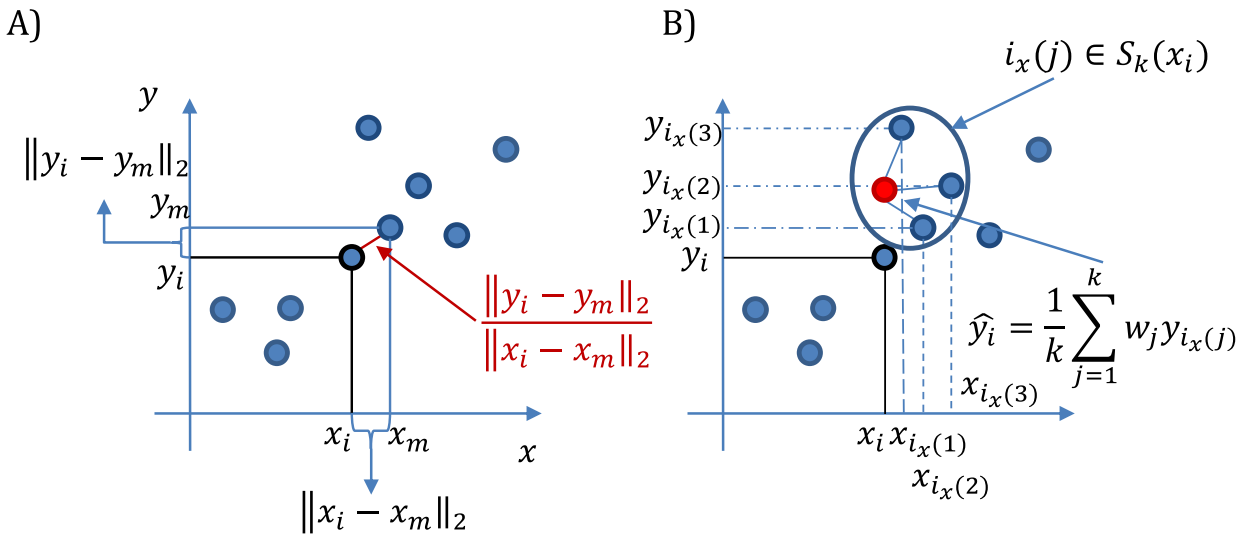


FIGURE 1. Graphical illustration of the Euclidean distance and *k*-NN algorithm. A) subfigure depicts the Euclidean distance between two points, while *k*-NN (B) evaluates the dependent variable based on the values of the closest (distance) neighbors in the independent variables.

performed to evaluate the accuracy of the predicted data. In the FNN algorithm, however, the closest neighbor is evaluated to determine whether it is located in the good or false neighbor set. A neighbor is considered a good neighbor if the mth point is the neighbor of the ith point in both independent and dependent variables $\{i(j) \in \mathrm{GNN}(\mathbf{x}_i) | i_x(j) \in S_1(\mathbf{x}_i) \wedge i_y(1) \in S_1(y_i)\}$ (depicted by green and red circles); however, one-way false neighbors may appear $\{i(1) \in \mathrm{FNN}(\mathbf{x}_i) | i_x(1) \in S_1(\mathbf{x}_i) \oplus i_y(1) \in S_1(y_i)\}$ (green or red circles). As such, points that are not neighbors of the ith point can also be defined as $\{i_x(1) \notin S_1(\mathbf{x}_i) \wedge i_y(1) \notin S_1(y_i)\}$ (blue circles).

The FNN can be modified for more than one neighbor, named the false k-nearest neighbor (Fk-NN). For a point with a neighborhood of k size:

$$F_k(\mathbf{x}_i, y_i) = \left\{ \frac{||y_i - y_{i_x(j)}||_2}{||\mathbf{x}_i - \mathbf{x}_{i_x(j)}||_2} > \alpha \middle| i_x(j) \in S_k(\mathbf{x}_i) \right\} \tag{16}$$

where the set $F_k(\mathbf{x}_i, y_i)$ denotes the set of false neighbors in a k member neighborhood. Note: the length of $F_k(\mathbf{x}_i, y_i)$ is not necessarily k.

Then the $F_k(\mathbf{x}_i)$ sets are calculated for each point, and their normalized length is taken as its mean:

$$\mathrm{Fk\text{-}NN} = 100 \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} |F_k(\mathbf{x}_i, y_i)| \tag{17}$$

The Fk-NN algorithm may be more robust than the FNN algorithm, as the underlying model is trained in local neighborhoods rather than the nearest neighbor.

D. TRUSTWORTHINESS AND CONTINUITY

Trustworthiness and Continuity (T&C) is another neighbor-based algorithm that measures the precision of projection from one set of variables to another [3].

The method counts the number of neighbors that are the k neighbors of the ith point in the other set of variables. T&C considers the degree of overlap between neighborhoods in both sets of variables, providing a metric of neighborhood similarity. The false nearest-neighbors technique is a special case of continuity. In this subsection, $i_y(j)$ denotes the indices of points that are the first kth neighbor of the ith point in the independent variables, while $i_x(j)$ represents the indices of points that are the first kth neighbor of the ith point in the independent variables.

Suppose that the rank in the independent variables is denoted by $r^x_{i,i_x(j)} \leq k$, then $\mathbf{x}_{i_x(j)}$ is the jth neighbor of $\mathbf{x}_i$. If $\mathbf{x}_{i_y(j)}$ cannot be located in the k neighborhood of $\mathbf{x}_i$, but $y_{i_y(j)}$ is in the k neighborhood of y_i, then a set of untrustworthy neighbors can be defined.

$$U_k(\mathbf{x}_i) = \left\{ i_y(j) \in \{1, \ldots, N\} \middle| r^y_{i,i_y(j)} \leq k \wedge r^x_{i,i_y(j)} > k \right\} \tag{18}$$

where $U_k(\mathbf{x}_i)$ denotes the set where the k number of neighbors of y_i are not considered the kth neighbors of $\mathbf{x}_i$.

The rank distances of the false neighbors from the neighborhood can be summed and scaled to the [0,1] interval, resulting in the trustworthiness measure.

$$T_k = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{i_y(j) \in U_k(\mathbf{x}_i)} (r^x_{i,i_y(j)} - k) \tag{19}$$

where T_k is the trustworthiness of the model based on kth neighbor local models, N is the number of observations, $U_k(\mathbf{x}_i)$ is the set of indices of neighbors in the dependent variable that are not the kth neighbors in the independent variables. Trustworthiness is also scaled to $0 \leq T_k \leq 1$ by $2/Nk(2N - 3k - 1)$ scaling coefficient.

Trustworthiness measures the accuracy of the projection from the dependent variables y_i to the independent variables $\mathbf{x}_i$. If the neighborhood of y_i is established and their neighbors are not included in the neighborhood in the independent variables ($\mathbf{x}_i$), then the projection is untrustworthy.

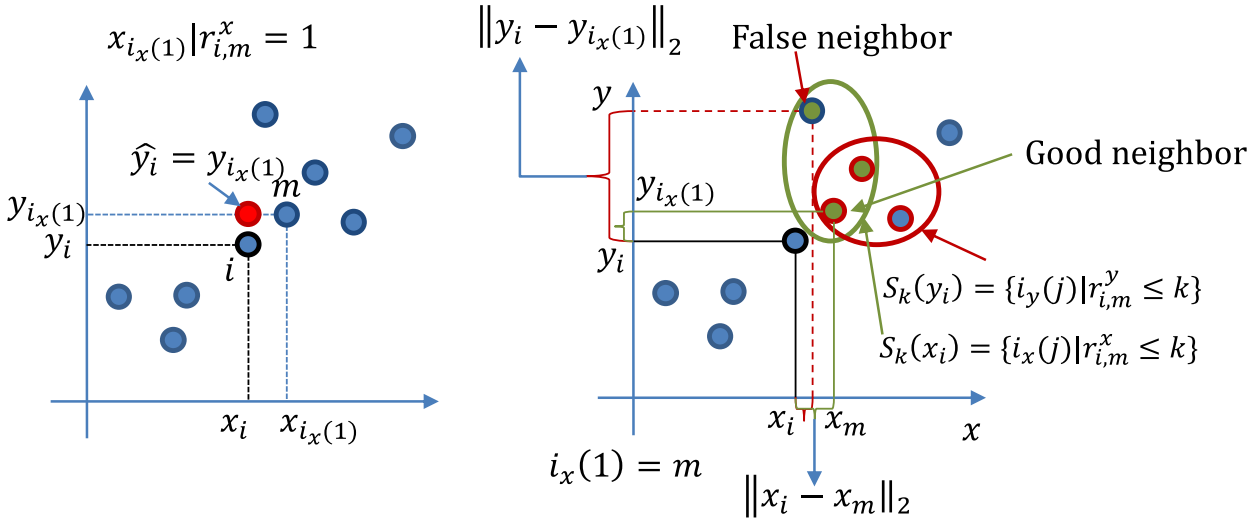


FIGURE 2. Graphical illustration of the nearest neighbors and false nearest neighbors methods. The left-hand side subfigure depicts the nearest neighbor method, where the closest neighbor determines the value of the dependent variable. The green and red circles depict the neighborhood in one variable (similarly illustrated with the red edges for the independent and green filling for the dependent). The FNN (B) algorithm evaluates the relationship between two variables by iterating through each point in the variables and determining which neighbors in one set of variables are neighbors in the others. If a neighbor can only be found in one, then it is only a false neighbor. However, if a point is located in both sets, it is considered a good neighbor (red edges and green filling).

Continuity measures the validity of projected points, where k neighbors $\mathbf{x}_{i_x(j)}$ may be located in the k neighborhood of $\mathbf{x}_i$, however, $r^y_{i,i_x(j)}$ is outside of the predefined range of k.

$$V_k(y_i) = \left\{ i_x(j) \in \{1, \ldots, N\} \middle| r^x_{i,i_x(j)} \le k \wedge r^y_{i,i_x(j)} > k \right\} \tag{20}$$

where $V_k(y_i)$ is the set where the neighbors of $\mathbf{x}_i$ are not neighbors of y_i

The continuity measure is formalized as:

$$C_k = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{i_x(j) \in V_k(y_i)} (r^y_{i,i_x(j)} - k) \tag{21}$$

where C_k is the continuity of the model based on kth neighbor local models, $V_k(y_i)$ is the set of neighbors that are not the kth neighbor in the dependent variable. Continuity is scaled to the range of $0 \le C_k \le 1$ by applying the $2/Nk(2N - 3k - 1)$ scaling coefficient.

Figure 3 presents the basic operation principle of trustworthiness and continuity measures. The upper axis denotes the indices of the closest neighbors in the dependent variable. The lower axis does the same for the independent variables. Both measures provide an evaluation of one-directional projection by the rank distance of the false neighbors from k-sized local neighborhoods. The indices denoted by green points are parts of the neighborhood in the dependent variables. At the same time, the red edge indicates membership in the local neighborhood in the independent variables. The ones with both marks are good neighbors, and those with only one are considered false neighbors, whose ranks are adjusted to their distance from the edge of the neighborhood. On the left-hand side, the trustworthiness measure is illustrated, which sums up the distance of the rank of the false neighbor from the local neighborhood in the independent variables and *vice versa* with continuity, but in the other direction. It is only reasonable to use the continuity measure for feature selection, as it evaluates the goodness of the projection of the first k neighbors in the independent variable to the dependent variable for each point.

E. NEIGHBORHOOD RANKING-BASED FEATURE SELECTION

Neighborhood ranking-based feature selection (NRFS) is the sum of the differences between the ranked distances of the independent and dependent variables.

When a combination of independent variables is selected, the Euclidean distance is calculated, which is then ranked, resulting in a matrix $\mathbf{R}^x_{N\times N}$. Similarly, the ranking is also established with the corresponding points of the dependent variable ($\mathbf{R}^y_{N\times N}$). Then the mean difference of the two matrices is calculated. It is crucial to scale the ranking differences to the zero-to-one interval, as the summation may lead to incomprehensible numbers.

The theoretical maximum of the Sum of Ranking Differences (SRD) method [4], [5], [6] is used as the scaling coefficient. It is reasonable that the ranking difference of the same vectors yields zero, whereas the maximum depends on the number of observations.

$$\mathrm{SRD}_{\max} = \begin{cases} 2\sum_{s=1}^{\frac{N}{2}} (2s-1) = 2\left(\frac{N}{2}\right)^2 & \text{if } N \text{ is even} \\ 2\sum_{s=1}^{\frac{N}{2}} 2s = 2\frac{N}{2}\left(\frac{N}{2}+1\right) & \text{if } N \text{ is odd} \end{cases} \tag{22}$$

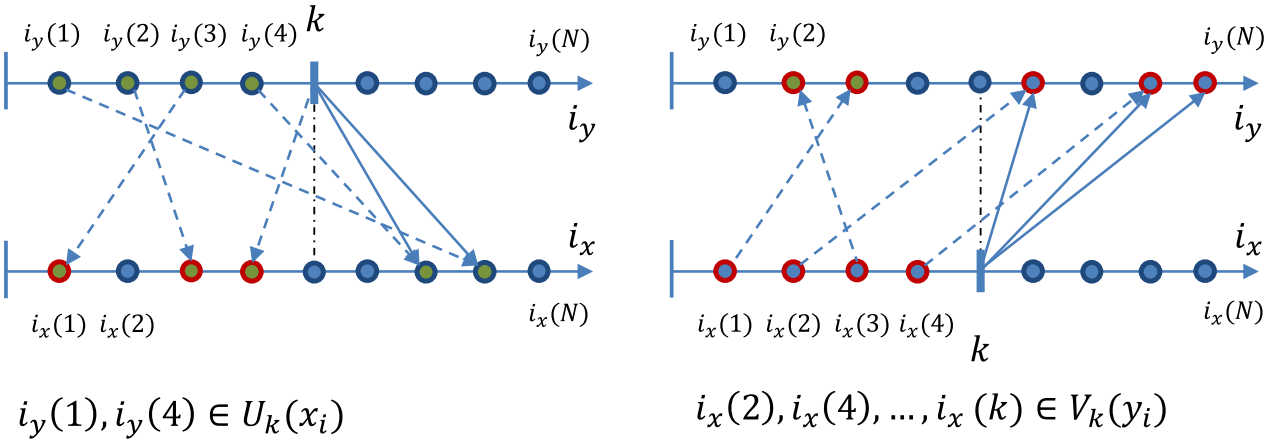


FIGURE 3. Graphical illustration of trustworthiness and continuity (T&C) methods. The left-hand side subfigure depicts the trustworthiness measure. Indices are sorted according to the closest neighbors in both variables, and the corresponding ranks are examined in the independent ones. Continuity is the opposite of trustworthiness. The right-hand side of the figure shows continuity. Moreover, points with a green filling or red edges are false neighbors, while having both is considered a sign of being a good neighbor. T&C can be calculated as in Eq. 19 and 21, respectively.

where s is the index of a rank. The theoretical maximum obtains its value by subtracting two opposite (in direction) rankings that facilitate N ranks. Generally, only half of the subtraction is required as the absolute values of the ranking differences are symmetric; only half is calculated if the number of observations is even. As an example, the ranking difference of $[1, 2, 3, 4]$ and $[4, 3, 2, 1]$ is $[-3, -1, 1, 3]$. This proposes that the ranking difference between two corresponding ranks is $(2s - 1)$, where s is the index of the ranks. A similar rule can be defined if the number of observations is odd. For example, there is the ranking difference between $[1, 2, 3, 4, 5]$ and $[5, 4, 3, 2, 1]$ that is $[-4, -2, 0, 2, 4]$ which can be generalized as $(2s)$. The rules only apply for one-half of the set. Thus, it is required to be multiplied by 2. If N is even, the sum of the differences returns $2(N/2)^2$, while if N is odd, the maximum ranking difference becomes $2(N/2) \cdot (N/2 + 1)$. Please see [5] for more information.

To evaluate the relationship between the independent and dependent variables, the neighbor ranks of the ith point in both sets of variables are subtracted, which is carried out in both ways:

$$\rho = 1 - \frac{\sum_{i=1}^{N}\sum_{m=1}^{N}(|r_{i,m}^{y} - r_{i,m}^{x}|)}{N\ \mathrm{SRD}_{\max}} \tag{23}$$

where ρ denotes the correlation of the independent and dependent variables, $r_{i,m}^{x}$ stands for the rank of the mth point against the ith point in the independent variables in $\mathbf{R}^x$. $r_{i,m}^{y}$ denotes the rank of the mth point against the ith data point according to the dependent variables. The ρ value can also be scaled to the $[0, 1]$ interval with the maximum SRD coefficient calculated in Eq. 22.

The NRFS can also be used with local neighborhoods in both independent (NRFSX) and dependent variables (NRFSY), similarly to T&C as the NRFS is considered its generalization. The first k ranked points are chosen in one set of variables, and the corresponding points in the other are subtracted.

With k being the number of neighbors, the set of indices is established:

$$i_x(j) \in I^x(r_{i,m}^{x} < k) \tag{24}$$
$$i_y(j) \in I^y(r_{i,m}^{y} < k) \tag{25}$$

where $i_x(j)$ denotes the indices of the k nearest neighbors in the independent variables.

The sets incorporate all kth nearest-neighbor indices. Thus, the local neighborhood-ranking-based feature selection can be defined as follows:

$$\rho_k^x = 1 - \frac{2\sum_{i=1}^{N}\sum_{i_x(j)\in I^x}(|r_{i,i_x(j)}^{x} - r_{i,i_x(j)}^{y}|)}{Nk(2N - 3k - 1)} \tag{26}$$
$$\rho_k^y = 1 - \frac{2\sum_{i=1}^{N}\sum_{i_y(j)\in I^y}(|r_{i,i_y(j)}^{x} - r_{i,i_y(j)}^{y}|)}{Nk(2N - 3k - 1)} \tag{27}$$

where ρ_k^x denotes the local variation of the NRFS algorithm in terms of the independent variables (NRFSX), while ρ_k^y stands for the local variation of the neighborhood ranking-based feature selection algorithm in terms of the dependent variables (NRFSY). Eq. 26 denotes the variant of NRFS in the neighborhood of the independent variable called NRFSX, while Eq. 27 defines the variant of NRFS in the neighborhood of the dependent variable, namely, NRFSY.

The method is a generalization of T&C as it provides information on the neighborhood difference in each set of variables.

F. DISCUSSION ON THE SIMILARITIES OF THE MEASURES

Throughout the paper, the relationship between the methods is stated. This section discusses the connection between the methods in depth.

The most obvious connection is between k-NN and FNN, which is the inequality case of the nearest neighbors, where

$k = 1$. As such, the error can be calculated as:

$$e_i = ||y_i - y_{i_x(1)}||_2 \tag{28}$$

The FNN algorithm is fundamentally based on the nearest neighbors, where the e_i error value only matters if it is more than a threshold value α and with equal weight:

$$e_i > \alpha ||\mathbf{x}_i - \mathbf{x}_{i_x(1)}||_2 \, |i_x(j) \in S_k(\mathbf{x}_i)| \tag{29}$$

Therefore, the FNN determines whether the distance quotient is above the threshold value. k-NN provides information on the validity of the model by taking the mean squared error value, while the FNN determines the validity of the model by summing up the number of false neighbors.

FNN is also related to the T&C technique, especially the continuity method. T&C focuses on the neighbors that are above a threshold that is specified in ordinal numbers. Therefore, the continuity measure is the ranking difference variant of the Fk-NN method.

Let us define the set of false neighbors:

$$F_k(\mathbf{x}_i, y_i) = \left\{ \frac{||y_i - y_{i_x(j)}||_2}{||\mathbf{x}_i - \mathbf{x}_{i_x(j)}||_2} > \alpha \middle| i_x(j) \in r^x_{i,i(j)} < k \right\} \tag{30}$$

The set is established based on the neighborhoods according to the first k ranks of the independent variables. We can also determine the false neighbors if the ranking difference is greater than the neighborhood. In other words, the rank cannot be found in the neighborhood of the dependent variable:

$$r^y_{i,i_x(j)} > k \middle| i_x(j) \in \{1, \ldots, N\} \tag{31}$$

We define the set of indices that are the kth nearest neighbors in the independent variables but are excluded from the neighborhood in the dependent variables:

$$V_k(\mathbf{x}_i, y_i) = \left\{ r^x_{i,i_x(j)} \leq k \wedge r^y_{i,i_x(j)} > k \middle| i_x(j) \in \{1, \ldots, N\} \right\} \tag{32}$$

The ratio of good neighbors can be calculated by one minus the false neighbors:

$$\texttt{Gk-NN} = 1 - \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} |F_k(\mathbf{x}_i)| \tag{33}$$

If the set of good neighbors is determined based on ranks, then the following is true:

$$\hat{C}_k = 1 - \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} |V_k(\mathbf{x}_i, y_i)| \tag{34}$$

The content of the set may provide more information on the local neighborhood, therefore, the distance of the ranks of the set from the neighborhood is summed up and scaled accordingly:

$$C_k = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{i_x(j) \in V_k(y_i)} (r^y_{i,i_x(j)} - k) \tag{35}$$

Both approaches measure the ratio of good neighbors. While the Gk-NN sums up the length of the set of good neighbors per point, continuity considers the amount of neighbors not in the k vicinity of the point. In this way, the rank deviation of the false neighbors is based on their distance from their neighborhood.

T&C technique can be defined as a special case of the novel local NRFS algorithms. Continuity only considers the rank of the neighbor if it is a false one, while the NRFSX calculates the total absolute difference of the dependent variable's ranks to the $[1, \ldots, k]$ ranking. This is also true for trustworthiness and NRFSY.

$$\rho^x_k = 1 - \frac{2 \sum_{m=1}^{N} \sum_{i_x(j) \in I^x} |r^y_{i_x(j),i} - r^x_{i_x(j),i}|}{Nk(2N - 3k - 1)}, \quad r^x_{i,i_x(j)} \in [1, 2, \ldots k] \tag{36}$$

Let us suppose that only the false neighbors are taken into account:

$$i_x(j) \in I^x(r^x_{i,m} \leq k \wedge r^y_{i,m} > k) = i_x(j) \in V_k(y_i) \tag{37}$$

Substituting into Eq. 36:

$$\hat{\rho}^x_k = 1 - \frac{2 \sum_{m=1}^{N} \sum_{i_x(j) \in V_k(y_i)} (|r^y_{i(j),m} - r^x_{i(j),m}|)}{Nk(2N - 3k - 1)} \tag{38}$$

If the ranking difference is measured from the end of the neighborhood:

$$C_k = 1 - \frac{2 \sum_{m=1}^{N} \sum_{i_x(j) \in V_k(y_i)} (|r^y_{i(j),m}| - k)}{Nk(2N - 3k - 1)}, \quad r^x_{i_x(j),m} > k \tag{39}$$

Thus, continuity is a special case of the NRFSX where the ranking differences from the neighborhood of the false neighbors are calculated.

The ranking difference is similar to the distance Spearman's correlation, as is a special case of the distance correlation (DC) technique [14]. The distance correlation measures the covariance and the variance between the multivariate variables. The significant difference from classic measures, such as Pearson, is that DC requires the Euclidean distance matrix of the variables to calculate the distance covariance and variance, similar to what is defined in Eq.1. The method can handle more than one independent and dependent variables.

First, the distance correlation takes the double-centered distances of the Euclidean distance matrix:

$$a_{i,m} = d^x_{i,m} - \bar{d}^x_{i,\cdot} - \bar{d}^x_{\cdot,m} + \bar{d}^x \tag{40}$$

where $a_{i,m}$ denotes the double-centered distance of the Euclidean distance of the ith and mth points in the independent variables. $\bar{d}^x_{i,\cdot}$ represents the mean of the ith row of the independent distance matrix, $\bar{d}^x_{\cdot,m}$ stands for the mean of the mth column, and $\bar{d}^x$ denotes the grand mean (mean of the means) of the independent distance matrix. Similarly to the distances in the independent variable, it can be done for the dependent as well:

$$b_{i,m} = d^y_{i,m} - \bar{d}^y_{i,\cdot} - \bar{d}^y_{\cdot,m} + \bar{d}^y \tag{41}$$

where $b_{i,m}$ stands for the double-centered distance of the Euclidean distance in the dependent variable.

The distance correlation is very much the same as the general Pearson correlation, with the exception that the (multivariate) Euclidean distance of the input data is examined instead of the original variables:

$$\mathrm{DC} = \frac{\sum_{i,m=1}^{N} a_{i,m} b_{i,m}}{\sqrt{\sum_{i,m=1}^{N} a^2_{i,m} \sum_{i,m=1}^{N} b^2_{i,m}}} \tag{42}$$

The ranking difference can be described as a particular case of distance correlation (DC). Let us suppose that the ranks are examined. If the ranking matrix consists of different ranks (assuming that no distance is the same), then a Spearman rank correlation based on the generalized correlation can be derived [15]:

$$\mathrm{RDC} = 1 - \frac{6 \sum_{i=1}^{N} \sum_{m=1}^{N} r^x_{i,m} - r^y_{i,m}}{N(N^2 - 1)} \tag{43}$$

The only difference between ranking difference and ranking distance correlation (RDC) is that the absolute value of the differences is taken, and thus the scaling coefficient is adjusted:

$$\rho = 1 - \frac{\sum_{i=1}^{N} \sum_{m=1}^{N} (|r^x_{i,m} - r^y_{i,m}|)}{N \mathrm{SRD}_{\max}} \tag{44}$$

where the $\mathrm{SRD}_{\max}$ denotes the equations described in Eq. 22.

G. RELATED WORKS OF NEIGHBORHOOD-BASED METHODS

Following the detailed description of the neighborhood based methods, the related works are thoroughly introduced to provide previously established use cases for the above-mentioned methodology and related methods.

There has been precedent for the use of neighborhoods in feature selection to improve classification performance. Neighborhoods can be analyzed with the help of entropy and select relevant features with low computational complexity [16]. Fuzzy neighborhood-based entropy methods measure the mutual information of input and output variables and aim to improve feature selection algorithms [17]. Moreover, the k nearest neighbors algorithm has been integrated into rough neighborhood-based feature selection algorithms [18]. Variable Neighborhood Search, which focuses on the optimum values of local neighborhoods, can also be used to select the optimal subset of variables [19]. The accuracy of the k-NN classification improves with other neighborhood-based feature selection [20], and being neighborhood-based itself, it may not require any other methods, except for an optimization algorithm to select features that provide optimal accuracy.

As a nonparametric regression model, k-NN [21], [22] has been researched as a potential feature selection algorithm, *e.g.*, k-NN classification has been used to accelerate feature selection [23]. Sequential Random k-nearest neighbors (SRk-NN) algorithm is used to select features based on the majority vote of nearest neighbor classifiers [24], and a distance- and attribute-weighted k-NN-based algorithm has also been utilized for feature selection [25]. The performance of k -NN-based feature selections has already been examined [26], and a method for tuning the number of neighbors (k) has also been developed, along with goal-oriented similarity measures [27]. There has been precedent for the use of genetic algorithm-based feature selection to improve the performance of k-NN in a classification problem [28]. In the Internet of Things application, the feature-selection aspect of k-NN has been utilized for the detection of network intrusion [29]. The k-NN algorithm is often modified, *e.g.* the differential nearest-neighbor regression approximates local gradients to evaluate n-th Taylor polynomial, and, therefore, replaces the mean function for prediction [30].

FNN is considered a method capable of supporting feature selection [31], and has been used to select the appropriate embedding dimension [12] and to detect determinism [32]. Practical applications include diagnosing bearing failure, where the FNN selects parameters that indicate malfunctioning operation [33]. Batteries have been analyzed by determining the minimum embedding dimensions, which are sent to a hybrid neural network to calculate the remaining lifetime [34]. Moreover, in near-infrared spectroscopy, FNN has been used to select characteristic wavelengths [35]. A mixture of FNN and Supervised Locality Preserving Projection (SLPP) has been applied to eliminate weak features based on a false neighborhood ratio [36].

Nonlinear correlation can also be assessed using the distance correlation method (DC) [14]. Similarly to neighborhood-based methods, DC requires distance matrices for both sets of variables. However, the significant difference is that the ranking is not used. The technique establishes correlation based on the Euclidean distance of the variables, whose distance covariance is divided by the product of the distance standard deviation. DC has been used successfully in feature selection, where DC feature selection performed well in synthetic datasets compared to other methods [37]. The technique has been tested in microarray classification problems that provide accurate models for class prediction [38].

In summary, while numerous methodologies implemented practical application of neighborhood-based methods in feature selection, the authors did not encounter publications with trustworthiness and continuity-based feature selection in its focus, neither a work that establishes the connections

between neighborhood-based methods. Therefore, this study aims to fill the research gap mentioned above.

III. RESULTS

Establishing a prediction model is not always expedient. It is essential to explore the information content of the dataset, as the model may not be able to reproduce the original output without the necessary input variables. Model-free methods can evaluate whether the dataset provides adequate information and whether the features are relevant to the output. These data-driven solutions may provide the necessary background for establishing models, although they require a selection algorithm. In simple cases where the number of variables is relatively small ($n <= 10$), the brute force approach can be used to find the correct combination of crucial features. In general, the methods are used as cost functions for optimization algorithms. As such, the proper interpretation of the evaluation scores must be discussed. The k-NN returns the mean squared error of the regression (which is to be minimized), FNN provides information on how many false neighbors can be found (and so it is to be minimized, albeit FNN returns zero to various combinations. In this case, the simplest solution should be selected. The remaining methods (NRFSX, Continuity (Cont), Distance correlation (DC), and rank distance correlation (RDC)) aim to demonstrate the correlation between the independent and dependent variables, which provides optimal solution at the maximum. During the brute force approach, minimum/maximum scores should be selected from the scores of all possible combinations. Its use is not advised if the dataset consists of more than ten features, as each combination is tested and can be time-consuming. Distance correlation and ranking distance correlation is described in Section II-F, with relevant equations (DC: 42, RDC: 43).

A possible selection method is the forward selection algorithm, which is a straightforward method for feature selection [7]. The method requires two sets, one of which is the set of unused variables, whereas the other contains the selected ones. The new feature is added to the set of selected features depending on which combination of new and previously selected variables produces the least error. The time complexity of this method is $O(n^2)$, while brute force requires $O\left(\sum_{c=1}^{n} \binom{n}{c}\right)$. Note that the order of selection is also provided, which may help establish an order of importance.

Performing feature selection with many ($n \gg 100$) features may decelerate the search, as both brute force and forward selection would require a lot of time to select the appropriate combination. As a response, heuristic algorithms were implemented to randomly choose combinations. A well-known example is the genetic algorithm (GA), whose variations are widely used in a similar context [8]. GAs are complex metaheuristic methods in which random samples aim to provide a sufficiently good solution. Iterative competition between samples continues for generations (iterations), which can result in the selection of a sufficiently performing combination of features. However, the correct combination may not be included in the starting population due to randomization. Therefore, GAs behave in a heuristic manner, but also retain some deterministic behavior in the selection process due to the iterative evaluation of feature combinations. As such, relevant variables often surface during the selection process, whereas irrelevant ones remain ''hidden''.

Selecting only a handful of individuals may be worth the effort, as the time required may drastically decrease. GAs are heuristic, and the evaluation of their performance should be statistical.

Neural networks are also included in this work, so the results are validated with a well-known and high-accuracy algorithm. If a valid model can be chosen, one should not forget that a model-driven structure is necessary. Neighborhood-based methods require metaparameters to be optimized (k, α), similar to neural networks (number of hidden layers).

Feature selection plays a vital role in identifying the order of dynamic systems. Here, the data required to calculate the output comprises the input variables and their time-shifted equivalents. The order of a system defines the lag required to accurately describe the system. Inputs can be systematically added to determine the order of the system; brute force can be used.

The next section presents the ability of the methods to describe nonlinear relationships through three simple equations. The Friedman dataset is also benchmarked with forward selection, GA, and validated by a neural network. Finally, we determine the order of a simulated polymerization reactor dataset as an example for dynamic system identification.

The main goal of this work is to establish a link between neighborhood-based methods and to use them for feature selection. The application potentials are discussed in the following subsections: In the first use case, we employ feature selection in linear, monotonous, and nonlinear equations. Then, we benchmark the methods on the Friedman data set [9] with forward selection, GA, and a neural network. Lastly, we analyze a simulated polymerization reactor to provide an example in which the model order is determined for the dynamic system models [39].

A. LINEAR, MONOTONOUS, AND PERIODIC FUNCTIONS

This subsection discusses didactic examples of feature selection problems. A four-feature dataset (n = 4) is generated for each problem, with a thousand uniformly random samples (N = 1000). The following simple equations represent linear, monotonic, and nonlinear problems.

$$y = x_1 + x_2 \tag{45}$$

$$y = x_1 + \log(x_2^2) \tag{46}$$

$$y = \cos(x_1) + \exp(x_2) \tag{47}$$

where x denotes the value of an independent variable, y stands for the value of the dependent variable

As the number of features is meager, the brute force approach is used for the selection algorithm. The following cost functions are selected: the k-NN, FNN, continuity, NRFSX, along with distance correlation, and rank distance correlation to benchmark against state-of-the-art methods. The number of nearest neighbors is selected to be ten for each relevant method, and no cross-validation was applied. The results are illustrated in Figure 4, where the scores of the combinations are values between zero and one, except for k-NN, where the mean square error is provided. Figure 4 is interpreted as follows: FNN sums up the inconsistency of neighbors in the dataset, and k-NN presents the mean squared errors of the incorporated features, therefore the scores must be minimized. The others aim to provide a correlation-like score and should be maximized. All approaches solve the feature selection problem accurately for linear and monotonic problems. The nonlinear equation is problematic for ranking and standard distance correlation.

B. FRIEDMAN-1 MODEL

The Friedman-1 data set is widely used in feature selection as it contains five relevant variables, including a nonlinear one, and five irrelevant random features [9]. A thousand uniformly random points are generated for each feature, between zero and one. The dependent variable is calculated as follows:

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \sigma \quad (48)$$

where x_i denotes a value of the ith independent variable, y stands for the value of the dependent variable and σ defines random noise.

The optimal number of neighbors must be determined before applying the methods. Therefore, we analyze the behavior of k-NN for one to fifty neighbors illustrated in Figure 5. As the database is generated randomly, the number of neighbors may differ with each retry. The seed of the random number generator was set to the same value for each experiment. In this example, the 12 nearest neighbors provided the most negligible error.

Neighborhood-based methods are used as cost functions for the forward selection optimizer algorithm. These methods are often considered to select features on a correlational basis and therefore, do not require cross-validation. Figure 6 illustrates the error of the methods and the order in which the features were selected.

The neighborhood-based methods each selected the relevant features in the same order. The error values of NRFSX, k-NN, FNN, and continuity had their optimums at the fifth inclusion of features, with the right combination of variables. Both distance correlations reached their optimum with the third. Nevertheless, DC determined the next two features, while RDC failed with the last. The MATLAB implementation of the rank-distance correlation is rather costly. It may fail due to the analysis of the entire ranking, which may not be able to represent a nonlinear correlation, such as that shown in Section III-A. Cross-validation has not been performed on neighborhood-based methods. Note that k-NN has utilized the leave-one-out correlation to evaluate feature relationship. It is a viable approach to cross-validate the neighborhoods and their accuracy with less data. If data is removed, the optimal number of neighbors will most probably change to cope with the removal of relevant points. We also included a way to calculate the k (nearest neighbors) hyperparameter which can be found in Figure 5.

We validate the methods against the neural network (NN), using the relevant features and all features as input data. The NN was built as a feedforward network with a hidden layer that contains 10 neurons. The predictive algorithm was also validated by a 10-fold cross-validation with a random partition, where the training (9 parts) and test (1 part) data were selected randomly for each evaluation. Figure 7 is included with one simple thing in mind; black-box algorithms have the innate ability to be applied as feature selection tools. The neural network (NN) is incorporated into a forward selection algorithm to directly compare with the performance of neighborhood-based methods. The mean squared errors are illustrated in the form of boxplots, where the NN performs better if any relevant features are added to the input set. If irrelevant features are chosen, however, cross-validation fails to provide constant results, as the method attempts to adjust to the noise. The boxplots represent the typical bias-variance problem, the validation error has a minimum at a given model complexity that relates to 5 features (4-2-1-5-3). As can be seen, when more features are added, the models become overtrained, so when all of the features are used (see the boxplot named 4-2-1-5-3-7-9-6-10-8) the median and the variation of the validation MSE error show statistically significant deterioration to what we registered at the selected five features. We believe that this example demonstrates the power of the proposed model-free feature selection algorithm, as the difference between the two cases is statistically significant. For further information on evaluating the improvement of the mean squared error in neural networks, see [40].

Genetic algorithms are also used as selection algorithms and are one of the best techniques for feature selection. The optimal number of neighbors must be determined before the GA is used. GAs are heuristic in nature; therefore, there is a chance that the optimal combination of features is not included in the starting population. We iteratively determined the combinations of variables needed to provide the least error and summarized them in Figure 8. The starting bitstring population was set to ten, while the algorithm lasted five generations with a 0.1 chance of mutation. The optimal number of neighbors remained at 12. The tests were performed ten times for each method.

It is hard to generalize because of the heuristic nature of GAs. In this case, the best method was the continuity method, which failed to determine only one variable once during the ten iterations. Other techniques also provided sufficient results. FNN was seemingly the weakest of each, as it cannot differentiate between good features combined with irrelevant ones. Although distance correlation (DC)

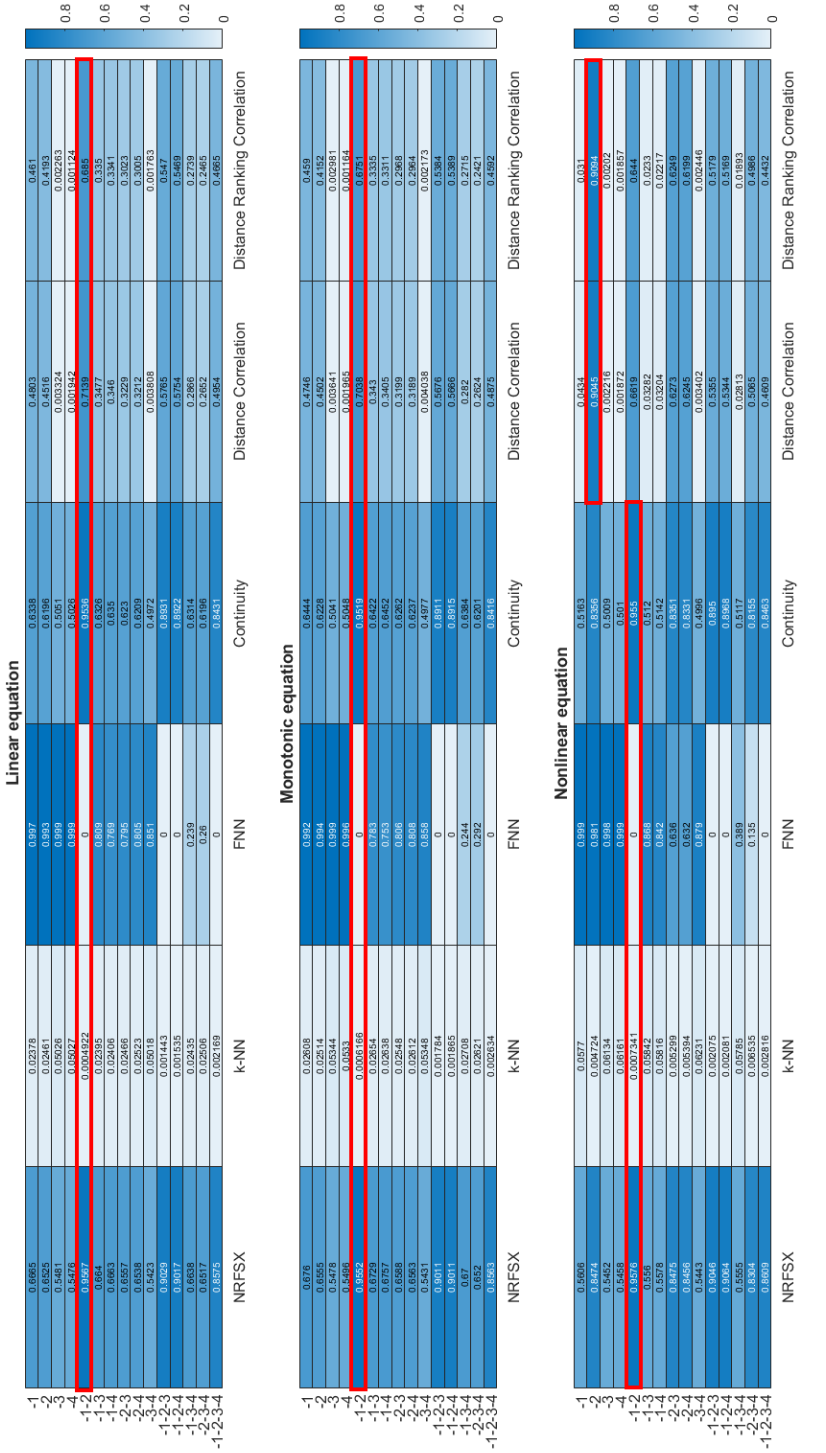


FIGURE 4. Heatmaps of the brute force feature selection algorithm with neighborhood-based methods as cost functions. The local neighborhood size is selected as ten (k = 10). *k*-NN and FNN cost functions should be minimized, while the others are maximized. Combinations with the best score are selected. In the figure, a red rectangle is drawn on top of the selected combination. The methods can undoubtedly solve the linear and monotonous functions, but the distance correlation (DC) and distance rank correlation (RDC) may fail against the nonlinear one. No cross-validation has been performed and all data have been included in the feature selection process. Cont denotes continuity.

from Eq. 42 and rank distance correlation (RDC) from 43 provided adequate results, we assume that GAs can select features with their help to some extent (as feature no. 3 was missed), however, we recommend *k*-NN and continuity for

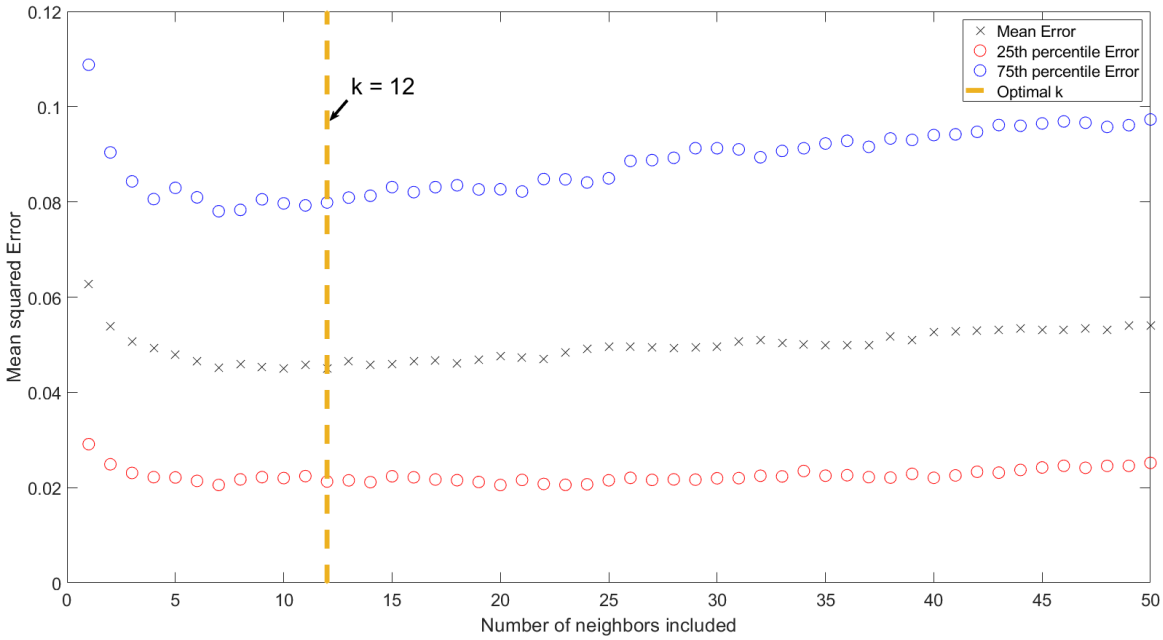


FIGURE 5. Selection of the optimal amount of neighbors. For each k, a regression is performed to search for the minimum mean squared error for the k-NN algorithm. The mean error yields the optimum if $k = 12$.

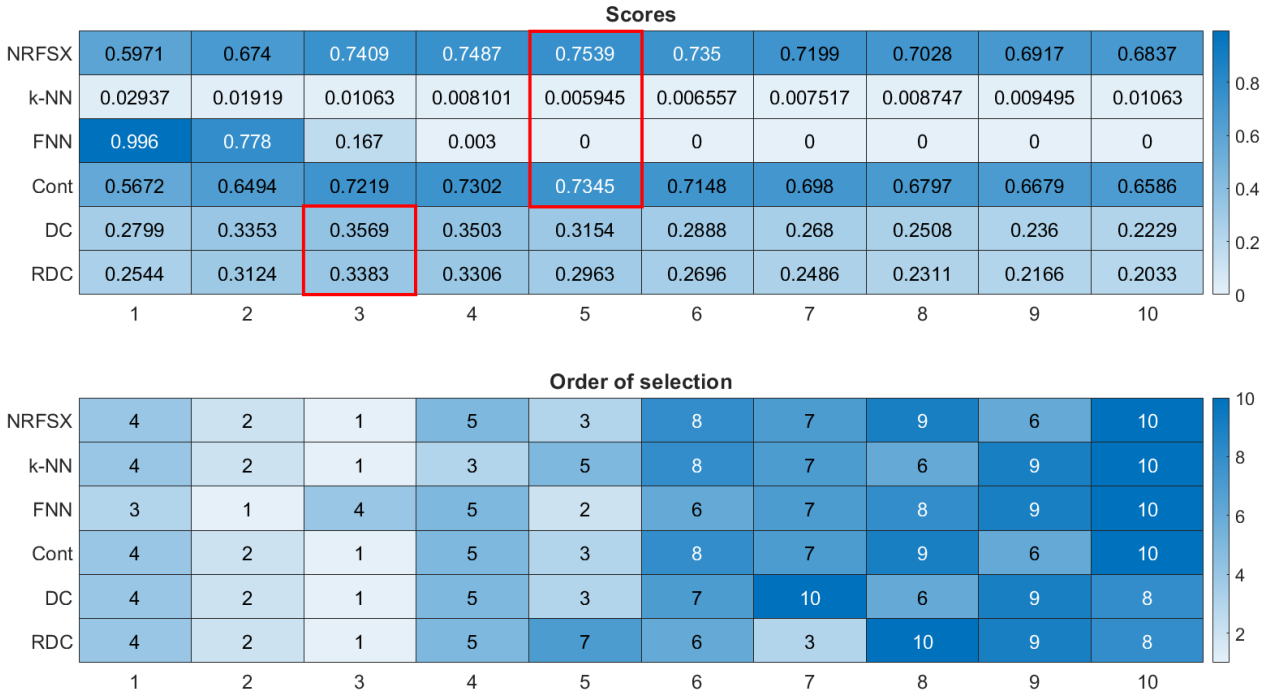


FIGURE 6. Forward selection with neighborhood-based methods as cost functions. The upper heatmap provides the scores (please note that for k-NN and FNN, a smaller score is better, while for the remaining scores, the higher the better). The optimums are framed with red rectangles. The lower table presents the order of the selected features. Cont denotes continuity.

GAs. We also measured the elapsed time for each method and iteration, which can be seen in Table 1. Although continuity (Cont) provided the best results, its runtime was three times the runtime of DC and twice the runtime of FNN. RDC failed to recognize two features properly. The two fastest were distance correlation and FNN, which selected more relevant features with shorter runtime. The higher runtime of NRFSX, continuity and RDC can be attributed to the additional ranking operation.

The neighborhood-based methods are capable of tackling the Friedman-1 dataset; however, other works proposed feature selection algorithms with different backgrounds that can solve the same problem. For example, graph-based feature selection may be able to build hierarchical models.

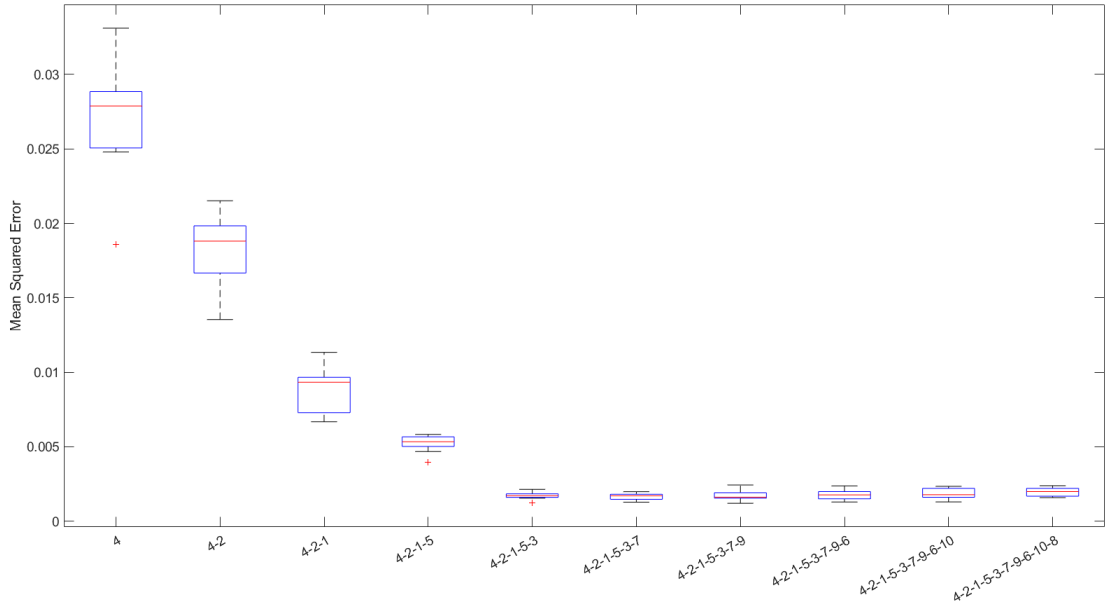


FIGURE 7. **The mean squared error of the neural network-based tenfold cross-validation for the Friedman dataset. A neural network with only the relevant features predicts the output with higher accuracy, whilst adding irrelevant variables increases the standard deviation of the mean squared error, due to the ability of the neural network to adjust for noise.**

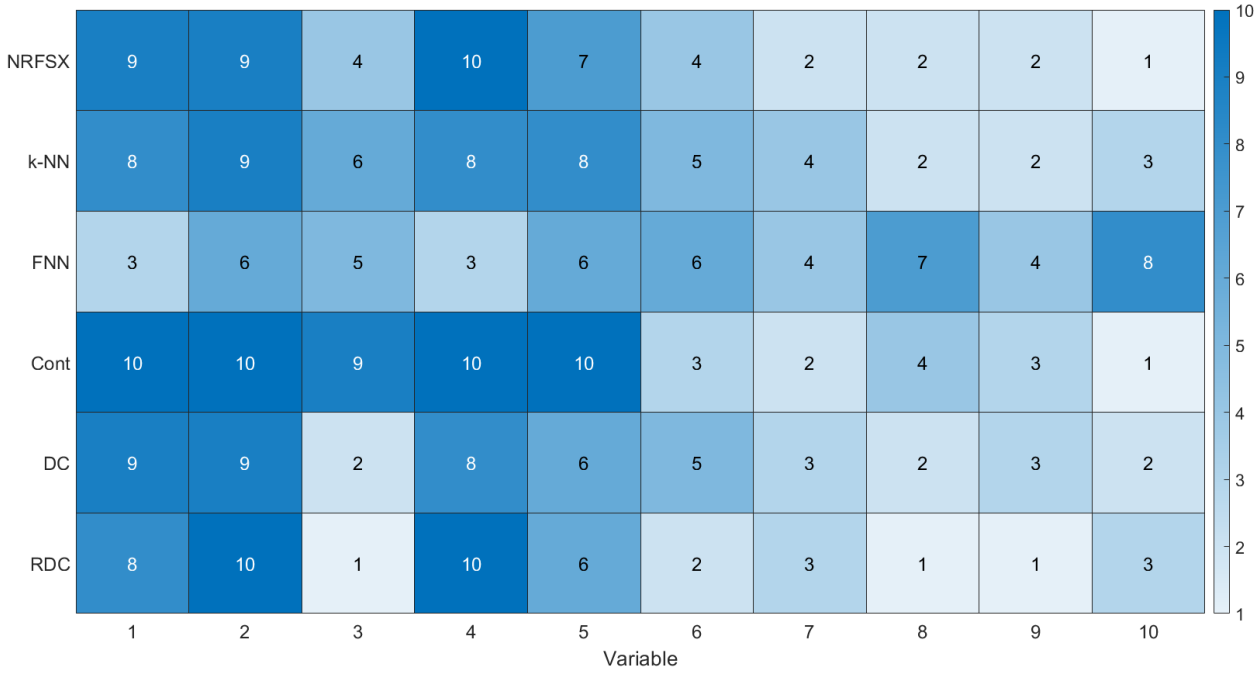


FIGURE 8. **Genetic algorithm with neighborhood-based methods as cost functions. Continuity (Cont) performed the best, while NRFSX and distance correlation (DC, Eq. 42) performed similarly. FNN failed, as the different combinations that include the correct variables still lack false neighbors, therefore, this approach cannot be used with a genetic algorithm.**

The best-path algorithm was tried on Friedman-1 while being validated by model accuracy metrics such as the Akaike information criterion [41]. However, the method focuses on mutual information and conditional relationships, and did not find a relevant variable, similar to the distance correlation. Indeed, neighborhood-based methods cannot yet build hierarchical models, but they are possible to implement. Variance-based decomposition is also a familiar algorithm in feature selection that proposes a low complexity technique with a solid mathematical background [42]. It aims to interpret the role of variables concerning the dependent variable that the neighborhood-based methods are not yet capable of; however, a SHAP-based approach may solve this issue [43]. Interpreting why the algorithm selects the feature is an essential aspect of the field; when using forward selection or brute force, there is often a ranking of features involved that may help establish which are dominant, or not as relevant as others with regard to the output variables.

TABLE 1. Genetic Algorithm runtime for each method.

Methods/Iteration	no. 1	no. 2	no. 3	no. 4	no. 5	no. 6	no. 7	no. 8	no. 9	no. 10	Mean
NRFSX	10.4769	10.2743	11.8696	10.5072	11.0044	10.5503	10.4024	9.4077	10.5933	10.3637	10.5450
k-NN	4.9930	4.9378	4.9488	4.6031	4.8697	4.4052	4.9572	4.9613	4.9510	4.5054	4.8132
FNN	3.9184	4.3142	5.2081	5.2904	5.2297	5.1969	5.2163	5.0848	5.1929	5.3659	5.0018
Continuity	9.3997	8.2149	8.0799	8.1588	8.0923	8.0539	8.0437	8.0926	8.0494	8.0377	8.2222
DC	2.9862	2.9760	3.0111	2.9308	2.9291	2.9126	2.9665	2.8030	2.9152	2.9023	2.9333
RDC	15.4333	15.3064	15.1424	15.3948	15.1941	14.4895	15.7480	15.7315	15.2093	15.6814	15.3331

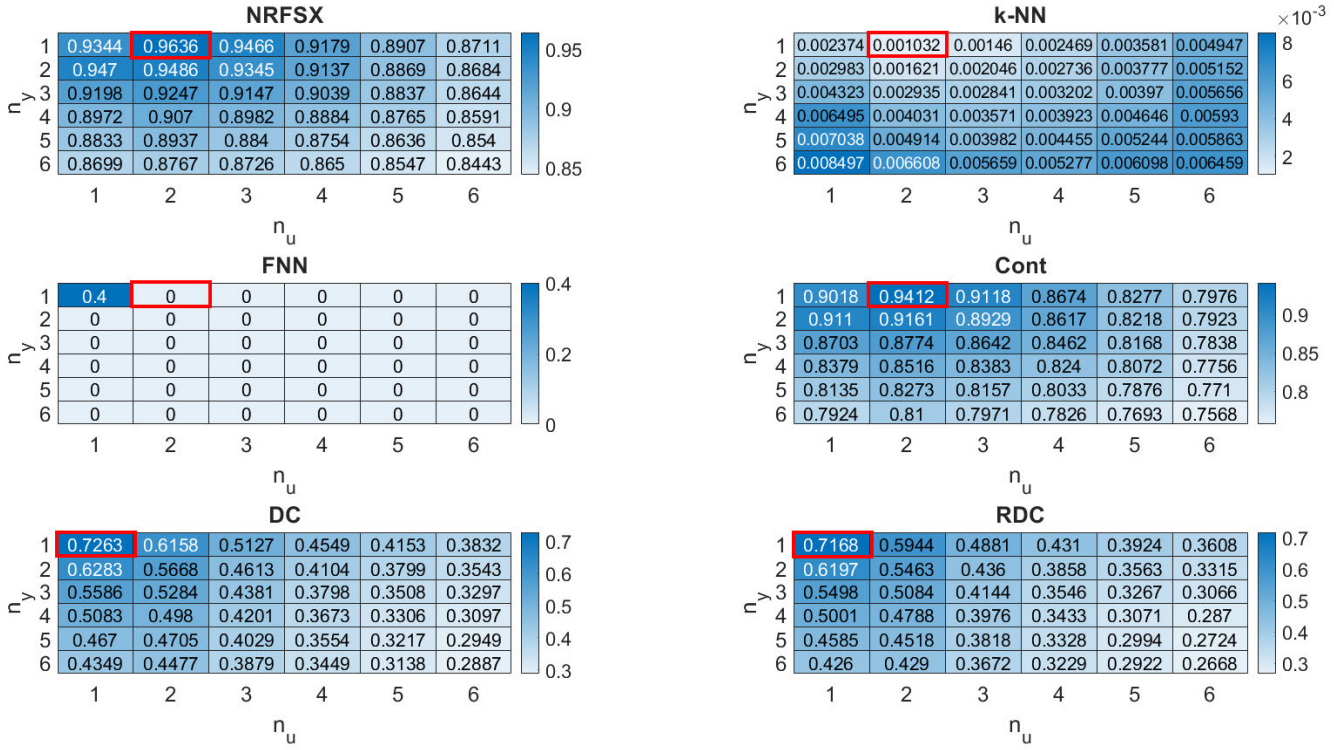


FIGURE 9. Order identification of the Polymerization reactor. Each neighborhood-based method can calculate the order of the reactor. DC and RDC may fail due to redundant input (time-delayed variable). The number of neighbors was selected to be $k = 3$.

C. DYNAMIC MODELING - POLYMERIZATION REACTOR

Dynamic modeling is a common task in process engineering. Identification of dynamic models may help boost the production of a factory; thus, it was selected as an application example for neighborhood-based techniques. The identification of reactor models provides an excellent use case for order identification. The dynamics within the reactor can be described with a nonlinear function f():

$$y_i = \mathrm{f}(\mathbf{x}_i);\ i = 1, \ldots, N \tag{49}$$

where y_i denotes the output, $\mathbf{x}_i$ stands for the input of the model at the ith point.

As the function is nonlinear, the nonlinear autoregressive models with eXogenous inputs (NARX) are employed for output prediction. This model predicts the output based on the past values of the input of the process (u_k) and the output (y_k). The order of the model is defined by the number of past values required for an accurate prediction [44]:

$$\mathbf{x}_i = [y_{i-1}, y_{i-2}, \ldots y_{i-n_y}, u_{i-1}, u_{i-2}, \ldots u_{i-n_u}] \tag{50}$$

where n_y denotes the order of the output and n_u stands for the order of the input.

We use uniformly random generated input between 0.005s and 0.015s using a simulation model of a continuous polymerization reactor by [39] that polymerizes methyl methacrylate with azobisisobutyronitrile (initiator) and toluene (solvent). A jacketed CSTR houses the reaction that can be analyzed, such as Eq. 50. For more information on this model, see [45]. The first six orders for both input and output were examined with brute force, using k-NN, FNN, NRFSX, continuity, distance, and rank distance correlations.

For the FNN, we use ratio tables to determine the order of the model. Identification works as follows: The first cell with a close to zero number closer to the left upper corner determines the order of the model [2]. In contrast, the others are minimized/maximized. The methods, except for DC and RDC, have been able to identify the order of the dynamic system, which is illustrated in Figure 9. Most were able to identify the system: NRFSX, k-NN, and continuity performed well to find a two-degree input and a one-degree output delay. In the FNN, a small number is delegated to ($n_u = 2, n_y = 1$).

DC fails because of the similarity between the variables, which are essentially the same vectors but time-delayed. The input of rank DC is also shifted, so the highest correlation will occur when only the first-order input and output are selected.

Two input delays ($n_u = 2$), and one output delay ($n_y = 1$) have also been identified for the polymerization reactor in [2], and the methods can identify the model order.

IV. CONCLUSION

This article demonstrated that neighborhood-based model-free feature selection can significantly improve data-driven modeling of complex nonlinear systems. Based on the integration of false neighbors and rank correlation, a novel method has been developed to select relevant variables and determine the correlation of the model variables. The analysis of the problem highlighted that FNN is the special case of k-NN regression with a leave-one-out validation as well as the special case of the continuity metric used to evaluate multidimensional embeddings.

The proposed metrics have been incorporated into brute-force, forward selection, and genetic feature selection algorithms. The test results obtained in the dynamic modeling of a polymerization reactor and in widely used benchmark data sets confirmed the applicability of the method.

In the future, we will examine how the developed tool can be applied for causality analysis, for outlier detection, and for evaluating regions where the data do not adequately cover the feature space. Similarly to feature selection, outlier detection or active learning will also be used iteratively to filter out invalid informative observations. As this work focuses heavily on regression-based feature selection, one can also perform feature selection for classification problems; however, this is outside the scope of the present work.

The benefit of using neighborhood-based methods is that it does not require model identification and evaluation. The black-box nature of the methods can also incorporate nonlinear correlation, whilst no model is defined. When one opts for complex systems analysis, building a model may not be worth the effort; the data may not be informative enough, or problems regarding identification may mask the relevancy of the input variables. This can be shown with neighborhood-based methods.

Any feature selection problem with enough data and information may be a potential use of the methods. For example, in the process of monitoring infrared spectroscopy, FNN has improved indexing [46], and an additional improvement in performance may include variable selection. T&C and k -NN-based topological mapping of infrared spectroscopy (TOPNIR) can widely be used in *e.g.* the petrol industry to predict the quality of the product [47], and could also be used to select relevant features for customer satisfaction and product development. The methods could also be useful in fault detection and how the variables become distorted by outliers. A similar work features dimensionality reduction with FNN and retains a high failure detection rate [48].

A disadvantage of the methods is that causality cannot be established and that there is no built-in interpretability. Additionally, the techniques require an optimization algorithm for the selection itself, as they are cost functions. It is important to note that the number of observed points directly influences the run-time of the neighbor-based algorithms. However, an immensely high observation count can significantly decelerate feature selection, for which data sampling may be a good solution.

REFERENCES

[1] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A, Gen. Phys.*, vol. 45, no. 6, pp. 3403–3411, Mar. 1992.

[2] B. Feil, J. Abonyi, and F. Szeifert, "Determining the model order of nonlinear input–output systems by fuzzy clustering," *Adv. Soft Comput.*, vol. 1, pp. 89–98, Jan. 2003.

[3] V. Onclinx, V. Wertz, and M. Verleysen, "Nonlinear data projection on non-Euclidean manifolds with controlled trade-off between trustworthiness and continuity," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1444–1454, Mar. 2009.

[4] K. Héberger, "Sum of ranking differences compares methods or models fairly," *TrAC Trends Anal. Chem.*, vol. 29, no. 1, pp. 101–109, Jan. 2010.

[5] K. Héberger and K. Kollár-Hunek, "Sum of ranking differences for method discrimination and its validation: Comparison of ranks with random numbers," *J. Chemometrics*, vol. 25, no. 4, pp. 151–158, Apr. 2011.

[6] K. Kollár-Hunek and K. Héberger, "Method and model comparison by sum of ranking differences in cases of repeated observations (ties)," *Chemometric Intell. Lab. Syst.*, vol. 127, pp. 139–146, Aug. 2013.

[7] K. Z. Mao, "Orthogonal forward selection and backward elimination algorithms for feature subset selection," *IEEE Trans. Syst., Man Cybern., B, Cybern.*, vol. 34, no. 1, pp. 629–634, Feb. 2004.

[8] X.-Y. Liu, Y. Liang, S. Wang, Z.-Y. Yang, and H.-S. Ye, "A hybrid genetic algorithm with wrapper-embedded approaches for feature selection," *IEEE Access*, vol. 6, pp. 22863–22874, 2018.

[9] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, Mar. 1991.

[10] M. Azadkia, "Optimal choice of k for k-nearest neighbor regression," 2019, *arXiv:1909.05495*.

[11] K. U. Syaliman, E. B. Nababan, and O. S. Sitompul, "Improving the accuracy of k-nearest neighbor using local mean based and distance weight," *J. Phys., Conf. Ser.*, vol. 978, Mar. 2018, Art. no. 012047.

[12] C. Rhodes and M. Morari, "Determining the model order of nonlinear input/output systems directly from data," in *Proc. Amer. Control Conf.*, vol. 3, Jun. 1995, pp. 2190–2194. [Online]. Available: https://ieeexplore.ieee.org/document/531288

[13] B. Feil, J. Abonyi, and F. Szeifert, "Model order selection of nonlinear input–output models—A clustering based approach," *J. Process Control*, vol. 14, no. 6, pp. 593–602, Sep. 2004.

[14] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *Ann. Statist.*, vol. 35, no. 6, pp. 2769–2794, Dec. 2007.

[15] C. Xiao, J. Ye, R. M. Esteves, and C. Rong, "Using Spearman's correlation coefficients for exploratory data analysis on big dataset," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 14, pp. 3866–3878, Sep. 2016.

[16] L. Sun, X. Zhang, Y. Qian, J. Xu, and S. Zhang, "Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification," *Inf. Sci.*, vol. 502, pp. 18–41, Oct. 2019.

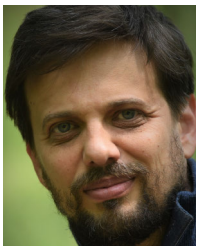
[17] L. Sun, M. Li, W. Ding, E. Zhang, X. Mu, and J. Xu, "AFNFS: Adaptive fuzzy neighborhood-based feature selection with adaptive synthetic oversampling for imbalanced data," *Inf. Sci.*, vol. 612, pp. 724–744, Oct. 2022.

[18] W. Xu, Z. Yuan, and Z. Liu, "Feature selection for unbalanced distribution hybrid data based on k-nearest neighborhood rough set," *IEEE Trans. Artif. Intell.*, vol. 5, no. 1, pp. 229–243, Jan. 2023.

[19] L. Matijević, "Variable neighborhood search for multi-label feature selection," in *Mathematical Optimization Theory and Operations Research*, P. Pardalos, M. Khachay, and V. Mazalov, Eds. Cham, Switzerland: Springer, 1007, pp. 94–107.

[20] M. A. N. D. Sewwandi, Y. Li, and J. Zhang, "A class-specific feature selection and classification approach using neighborhood rough set and K-nearest neighbor theories," *Appl. Soft Comput.*, vol. 143, Aug. 2023, Art. no. 110366.
[21] G. Guo, D. Neagu, and M. T. D. Cronin, "Using kNN model for automatic feature selection," in *Pattern Recognition and Data Mining*, S. Singh, M. Singh, C. Apte, and P. Perner, Eds. Berlin, Germany: Springer, 2005, pp. 410–419.
[22] D. A. Adjeroh, E. J. Harner, and S. Li, *Random KNN Modeling and Variable Selection for High Dimensional Data*. Morgantown, WV, USA: West Virginia Univ., 2009.
[23] A. Wang, N. An, G. Chen, L. Li, and G. Alterovitz, "Accelerating wrapper-based feature selection with K-nearest-neighbor," *Knowl.-Based Syst.*, vol. 83, pp. 81–91, Jul. 2015.
[24] C. H. Park and S. B. Kim, "Sequential random k-nearest neighbor feature selection for high-dimensional data," *Exp. Syst. Appl.*, vol. 42, no. 5, pp. 2336–2342, Apr. 2015.
[25] P. Bugata and P. Drotár, "Weighted nearest neighbors feature selection," *Knowl.-Based Syst.*, vol. 163, pp. 749–761, Jan. 2019.
[26] M. Pal, T. B. Charan, and A. Poriya, "*K*-nearest neighbour-based feature selection using hyperspectral data," *Remote Sens. Lett.*, vol. 12, no. 2, pp. 132–141, Feb. 2021.
[27] R. Puspadini, H. Mawengkang, and S. Efendi, "Feature selection on K-nearest neighbor algorithm using similarity measure," in *Proc. 3rd Int. Conf. Mech., Electron., Comput., Ind. Technol. (MECnIT)*, Jun. 2020, pp. 226–231.
[28] N. Maleki, Y. Zeinali, and S. T. A. Niaki, "A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113981.
[29] M. Mohy-eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection," *Multimedia Tools Appl.*, vol. 82, no. 15, pp. 23615–23633, Feb. 2023.
[30] Y. Nader, L. Sixt, and T. Landgraf, "DNNR: Differential nearest neighbors regression," in *Proc. 39th Int. Conf. Mach. Learn.*, vol. 162, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., 2022, pp. 16296–16317.
[31] A. Krakovská, K. Mezeiová, and H. Budáč ová, "Use of false nearest neighbours for selecting variables and embedding parameters for state space reconstruction," *J. Complex Syst.*, vol. 2015, pp. 1–12, Mar. 2015.
[32] R. Hegger and H. Kantz, "Improved false nearest neighbor method to detect determinism in time series data," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 60, no. 4, pp. 4970–4973, Oct. 1999.
[33] S. Zuo and Z. Liao, "Bearing fault dominant symptom parameters selection based on canonical discriminant analysis and false nearest neighbor using GA filtering signal," *Math. Problems Eng.*, vol. 2020, pp. 1–13, Apr. 2020.
[34] G. Ma, Y. Zhang, C. Cheng, B. Zhou, P. Hu, and Y. Yuan, "Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network," *Appl. Energy*, vol. 253, Nov. 2019, Art. no. 113626.
[35] Q.-P. Mei, T.-F. Li, L.-Z. Yao, X.-H. Liu, Y.-L. Hu, and L. Hu, "Characterization of a wavelength selection method using near-infrared spectroscopy and partial least squares with false nearest neighbors and its application in the detection of the chemical oxygen demand of waste liquid," *Spectrosc. Lett.*, vol. 52, no. 9, pp. 553–562, Oct. 2019.
[36] X.-H. Gu, T.-F. Li, L.-P. Yang, J. Yi, and W. Zhou, "Original feature selection based on false nearest neighbor criterion in supervised locality preserving subspace," *Opt. Precis. Eng.*, vol. 22, no. 7, pp. 1921–1928, 2014.
[37] H. Tan, G. Wang, W. Wang, and Z. Zhang, "Feature selection based on distance correlation: A filter algorithm," *J. Appl. Statist.*, vol. 49, no. 2, pp. 411–426, Jan. 2022.
[38] A. Brankovic, M. Hosseini, and L. Piroddi, "A distributed feature selection algorithm based on distance correlation with an application to microarrays," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 6, pp. 1802–1815, Nov. 2019.
[39] C. Rhodes and M. Morari, "Determining the model order of nonlinear input/output systems," *AIChE J.*, vol. 44, no. 1, pp. 151–163, Apr. 2004.
[40] U. Ahmed, A. R. Khan, S. Razzaq, and A. Mahmood, "Comparison of memory-less and memory-based models for short-term solar irradiance forecasting," in *Proc. 7th Int. Multi-Topic ICT Conf. (IMTIC)*, Jamshoro, Pakistan, May 2023, pp. 1–6.
[41] L. Riso, M. G. Zoia, and C. R. Nava, "Feature selection based on the best-path algorithm in high dimensional graphical models," *Inf. Sci.*, vol. 649, Nov. 2023, Art. no. 119601.
[42] F. Kamalov, "Orthogonal variance decomposition based feature selection," *Expert Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115191.
[43] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 4–10.
[44] S. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Hoboken, NJ, USA: Wiley, Jul. 2013.
[45] F. J. Doyle, B. A. Ogunnaike, and R. K. Pearson, "Nonlinear model-based control using second-order Volterra models," *Automatica*, vol. 31, no. 5, pp. 697–714, May 1995.
[46] J. Abonyi, T. Kulcsár, G. Sárossy, G. Bereznai, and R. Auer, "Visualization and indexing of spectral databases," *World Acad. Sci., Eng. Technol.*, vol. 67, pp. 147–152, Jan. 2012.
[47] T. Kulcsár, G. Bereznai, G. Sárossy, R. Auer, and J. Abonyi, "Visualisation of high dimensional data by use of genetic programming: Application to on-line infrared spectroscopy based process monitoring," *Soft Comput. Ind. Appl.*, vol. 223, pp. 223–231, Nov. 2014.
[48] J. Yi, L. Wu, W. Zhou, H. He, and L. Yao, "A sparse dimensionality reduction approach based on false nearest neighbors for nonlinear fault detection," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 4980–4992, Aug. 2021.

ÁDÁM IPKOVICH received the bachelor's degree in mechatronics engineering, in 2023. His research interests include rankings, multi-criteria decision-making and analysis, model interpretability, and many-objective optimization. His current research focuses on climate change, sustainability, and complex systems.



JÁNOS ABONYI received the M.Eng. and Ph.D. degrees in chemical engineering from the University of Veszprem, Hungary, in 1997 and 2000, respectively, and the Habilitation degree in process engineering and the D.Sc. degree from the Hungarian Academy of Sciences, in 2008 and 2011, respectively. From 1999 to 2000, he was with the Control Laboratory, Delft University of Technology, The Netherlands. He is currently a Full Professor in computer science and chemical engineering with the Department of Process Engineering, University of Pannonia. He has coauthored more than 250 journal articles and chapters in books. He has published five research monographs and one Hungarian textbook about data mining. His research interests include complexity, process engineering, quality engineering, data mining, and business process redesign.